MSX

truuks en tips deel 3

Hans Klopper en Marcel Le Belle



MSX truuks en tips deel 3 Hans Klopper en Marcel Le Belle

MSX
Truuks en tips
Deel 3

MSX

truuks en tips

deel 3

Hans Klopper en Marcel Le Belle

uitgeverij STARK - TEXEL
postbus 302 - 1794 ZG Oosterend tel. 02223 - 661

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

MSX

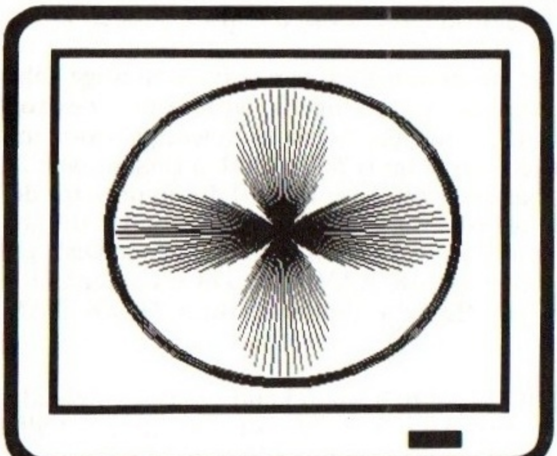
MSX truuks en tips –Oosterend : Stark-Texel
Dl.3 Hans Klopper en Marcel le Belle.
ISBN 90-6398-910-5
SISO 365.3 UDC 681.3.06
Trefw.: programmeren (computer) / MSX (computer).

1e druk 1986
ISBN 90 6398 405 7

Inhoud

pag.

Inleiding



In navolging van truuks & tips deel 1/2, heeft u hier het verwachte "MSX TRUUKS & TIPS DEEL 3". Dit derde deel uit de reeks bevat naast enkele eenvoudige BASIC truuks, zoals u gewend was, ook de nodige machinetaal routines. Logisch, gezien het feit, dat dit boek geschreven is door de auteurs van het MSX machinetaal handboek. Kennis van machinetaal is niet vereist om de programma's te kunnen toepassen. We zullen echter voor de volledigheid wel een korte inleiding in machinetaal geven. Voor degene die het MSX machinetaal handboek reeds in hun bezit hebben, zijn boven genoemde programma's zeer leerzaam.
De meeste routines zijn opgenomen in een programma, waardoor de toepassingsmogelijkheden wat duidelijker naar voren komen. We zijn er van overtuigd dat het merendeel van de programma's echt uniek zijn en dat u ze nog nergens anders heeft aangetroffen. Dit maakt MSX truuks & tips deel 3 tot een onmisbaar onderdeel van uw MSX boeken kollektie.

Een van de programma's is een softwarematige geheugenuitbreiding, waardoor het mogelijk wordt om twee programma's tegelijkertijd in het geheugen te bewaren. Door middel van een nieuwe instruktie (CMD) kunt u tussen twee 32K RAM geheugenbanken switchen. U heeft dus in principe de beschikking over het volledige RAM geheugen onder BASIC. Omdat we echter rekening hebben gehouden met Disk gebruikers, blijft er netto 2*24K RAM over. Dit is echter aan te passen, zodat de cassette gebruiker over ruim 2*28K RAM kan beschikken!

Ook BASIC leent zich uitstekend voor handigheidjes bij het programmeren. Zo kunt u in dit boek een in BASIC geschreven routine terug vinden die, indien twee sprites tegen elkaar gebotst zijn, uitzoekt welke sprites er zijn gebotst. Dit in

tegenstelling tot het ON SPRITE GOSUB kommando, welke alleen 'ziet' dat er twee sprites gebotst zijn, maar niet welke.

Naast bovenstaande geavanceerde programma's, kunt u ondermeer in dit boek terug vinden:

– Softwarematige reset, voor de MSX gebruikers die het zonder RESET knop moeten stellen.

– De "OK" prompt veranderen in elke gewenste tekst.

– Het razend snel uitwisselen van twee grafische schermen.

– Het kunnen uitlezen van b.v. de SELECT toets.

– Truuks met ROM routines

enz. enz. enz.

Zoals u ziet is in dit boekje voor iedere MSX gebruiker wel iets te vinden. We hopen dat de routines u verder kunnen helpen met het ontwikkelen van programma's op uw MSX computer.

Onze dank gaat uit naar ELECTRONICS NEDERLAND voor het beschikbaar stellen van twee SPECTRAVIDEO X' PRESS computers met het Wordstar tekstverwerkings programma waarmee we onze teksten geschreven hebben.

Tot ziens in deel vier.

H. Klopper & M. le Belle

Goochelen met tekst

Heeft u ooit behoefte gehad aan speciale lettervormen? Met onderstaand programma kunt u allerlei soorten letters kreëren. De tekst die u links bovenaan het scherm plaatst, zal op verschillende plaatsen en manieren elders op het scherm verschijnen.

In regel 30 wordt de tekst gedefinieerd die we willen gebruiken om de grappen mee uit te halen.

In regel 50 t/m regel 100 worden verschillende plaatsen en effekten op het scherm aangegeven. De gegevens in deze regels kunt u veranderen, zodat er iets ontstaat, dat aan uw eigen wensen voldoet. Met de "GOSUB 120" wordt naar de routine gesprongen die zorgt voor de eigenlijke akties.

In regel 140 worden door middel van het POINT-statement de coördinaten uitgelezen.

x-coördinaten: 0 - 100
y-coördinaten: 0 - 7

Dus in totaal worden er 8*101=808 puntjes uitgelezen. De gehele procedure zal daarom enige tijd vergen, maar het resultaat is zeer de moeite van het aanschouwen waard.

```
1 REM ***************************
2 REM *                         *
3 REM * Dit programma creëert   *
4 REM * verschillende vormen    *
5 REM * van de linksboven in het*
6 REM * scherm staande tekst.   *
7 REM *                         *
```

```
8 REM ***************************
9 :
10 A=0:OPEN"grp:"AS1
20 COLOR 1,14,14:SCREEN2,0:DEFINTA-Z
30 A$="TRUUKS & TIPS 3"
40 FOR X=1TO LEN(A$):A=A+6:PRESET(A,0):P
RINT#1,MID$(A$,X,1):NEXT
50 DEFFNX=240-Y:DEFFNY=X:GOSUB120
60 DEFFNX=250-X:DEFFNY=190-Y:GOSUB120
70 DEFFNX=12+X-Y:DEFFNY=12+Y+X:GOSUB120
80 DEFFNX=50+X*2:DEFFNY=140+Y*2:GOSUB120
90 DEFFNX=100+X-Y:DEFFNY=12+Y*4:GOSUB120
100  DEFFNX=120+X-Y/8:DEFFNY=62+Y*3+5:GO
SUB120
110 GOTO 110
120 FORX=0TO100
130 FORY=0TO7
140 DOT=POINT(X,Y)
150 IFDOT=0THEN170
160 PSET(FNX,FNY),DOT
170 NEXTY,X
180 RETURN
```

Kleuren instellen

Veel mensen hebben problemen met het vinden van een juiste instelling van hun kleuren op een TV/Monitor. Voor deze mensen is het volgende programma ideaal. Er worden allereerst verschillende kleurenbalken vertikaal naast elkaar geplaatst. Hierna worden er na een druk op de spatiebalk horizontaal enkele balken op het scherm afgebeeld. Als u dan nogmaals op de knop drukt, zullen er allemaal gekleurde blokjes op het scherm verschijnen. Als elke kleur goed is te onderscheiden van de andere en er geen noemenswaardige interferentie tussen de verschillende kleuren optreedt, dan is uw Monitor/TV goed afgesteld.

```
1 REM ****************************
2 REM *                          *
3 REM * Instellen van kleuren.   *
4 REM *                          *
5 REM ****************************
6 :
10 DIM A(31) : FOR I=1 TO 15
20 A(I)=I : A(I+16)=I : NEXT I
30 SCREEN2 : COLOR 15,1,1
40 FOR I=0 TO 15
50 LINE(I*16,0)-(I*16+15,191),I,BF
60 NEXT
70 X$=INKEY$ : IF X$="" THEN 70
80 FOR I=0 TO 15
90 LINE(0,I*12)-(255,I*12+11),I,BF
100 NEXT
110 X$=INKEY$ : IF X$="" THEN 110
120 FOR I=0 TO 15
130 FOR J=0 TO 15
```

```
140 LINE(I*16,J*12)-(I*16+15,J*12+11),A(
J+I),BF
150 NEXT J,I
160 X$=INKEY$ : IF X$="" THEN 160
170 GOTO 40
```

Diskette tester

Wilt u er zeker van zijn dat uw diskettes in orde zijn, dan is deze utility voor u zeker de moeite van het intikken waard. Er wordt nl. gekeken of alle sectoren van de schijf nog wel goed zijn. Als dat zo is dan wordt voor elke sector apart de boodschap "verify OK" gegeven.

Is de drive om een of andere rede niet in staat om een sector te lezen, dan wordt de boodschap "Disk error in sector..." gegeven.

Het uitlezen van een sector wordt gedaan in regel 90 door middel van de funktie DSKI$.

Syntax: DSKI$ (D,N)

De "D" staat voor de gebruikte drive.
De "N" staat voor de te lezen sector.

```
1 REM ***************************
2 REM *                         *
3 REM * Disk controle programma *
4 REM *      version 1.1        *
5 REM *                         *
6 REM ***************************
7 :
10 COLOR 15,4:SCREEN0:WIDTH38:KEYOFF
20 CLEAR 1000
30 INPUT"Hoeveel sectoren (369/719)";SEC
T
40 IF SECT>719 THEN PRINT"Te veel sector
en":FOR T=1 TO 750:NEXT:CLS:GOTO 30
50 IF SECT=0 THEN END
```

```
60 SECT=SECT-1:CLS
70 FOR N=0 TO SECT
80 ON ERROR GOTO 140
90 A$=DSKI$(D,N)
100 LOCATE 5,5:PRINT "Sector ";N;"verify
 OK"
110 ON ERROR GOTO 140
120 NEXT
130 END
140 PRINT"Disk error in sector ";N
150 N=N+1:RESUME
```

Cirkel tekenen

Als u een cirkel tekent vanuit BASIC dan denkt u waarschijnlijk niet direkt hoe de computer met door u ingevoerde gegevens (middelpunt, straal en vorm) een berekening uitvoert. Deze berekening vindt plaats door middel van een routine uit het ROM geheugen van uw computer. Deze routine komt ongeveer overeen met degene die gebruikt wordt in onderstaand programma.
U ziet dat u het als MSX gebruiker toch wel ontzettend makkelijk heeft. Veel andere computers moeten het stellen zonder een cirkel kommando! Stelt u zich dat eens voor.

```
1 REM ***************************
2 REM *                         *
3 REM * Het tekenen van een     *
4 REM * cirkel d.m.v. gonio     *
5 REM * metrie.                 *
6 REM *                         *
7 REM ***************************
8 :
10 LOCATE,,0
20 DIMX(315),Y(315)
30 FOR T=0 TO 6.28 STEP .02:K=K+1
40 Y(K)=SIN(T):X(K)=COS(T)
50 LOCATE10,5:PRINTK
60 NEXT
70 LOCATE,,1
80 COLOR 15,4,4:SCREEN2
90 FOR K=1TO315
100 PSET(X(K)*30+80,Y(K)*40+80):NEXT
110 GOTO110
```

Karakterset 90 graden draaien

Het volgende programmaatje draait de karakterset 90 graden met de klok mee.

Het programma kan als onderdeel van een ander programma funktioneren.

Met een willekeurig SCREEN kommando wordt alles weer ongedaan gemaakt.

P.S. Nadat u het programma eenmaal heeft gerund, kunt u het SCREEN1 kommando verwijderen. Run het programma nu nogmaals en verbaast u zich over het verkregen resultaat...

```
1 REM ***************************
2 REM *                         *
3 REM * Karakterset 90° draaien *
4 REM *                         *
5 REM ***************************
6 :
10 SCREEN1
20 LOCATE2,5:PRINT"ABCDEFGHIJKLMNOPQRSTU
VWXYZ"
30 LOCATE2,6:PRINT"abcdefghijklmnopqrstu
vwxyz"
40 CLEAR 200,&HE400 :DEFINT A-Z
50 FOR X=ASC("A")*8 TO ASC("z")*8+8
60 A=VPEEK(X):Q=Q+1
70 POKE&HE3FF+Q,A
80 NEXT X
90 FOR V=0 TO 58*8 STEP 8
100 FOR MM=7 TO 0 STEP-1
```

```
110 FOR X=0 TO 7
120 A=PEEK(&HE400+X+V)
130 B=(A OR (2^MM) )
140 IF B=A THEN GOSUB180
150 NEXT:VPOKEASC("A")*8+TT,FF:FF=0:TT=T
T+1:NEXTMM
160 NEXT V
170 END
180 FF=(FF OR (2^X) ):RETURN
```

Computerbank

Het nu volgende programmaatje geeft u in no time een overzicht van de in een bedrag voorkomende geldeenheden.

Het laat duidelijk zien dat een computer, mits goed geprogrammeerd, zeer snel in staat is om lastige klusjes te klaren.

```
1 REM ***************************
2 REM *                         *
3 REM *     Computer BANK       *
4 REM *                         *
5 REM ***************************
6 :
10 SCREEN0:WIDTH40:KEYOFF
20 LOCATE12,2:PRINT"COMPUTER BANK"
30 LOCATE12,3:PRINT"-------------"
40 BR$="biljet(ten) van "
50 LOCATE0,8:PRINT"--------------------
--------------------"
60 LOCATE8,6
70 LOCATE 0,10:PRINT"-Geen centen, want
die zijn uit de rou-":PRINT" latie!"
80 LOCATE 0,13:PRINT"-Guldens en centen
worden gescheiden":PRINT" door een punt
i.p.v. een komma.
90 LOCATE 0,5:INPUT "Geef een bedrag op"
;A
100 CLS
110 LOCATE0,5,0:PRINT"Het bedrag f";A"be
vat:
120 D=INT(A/1000)
130 H=INT(A/100-D*10)
```

```
140 T=INT(A/10-D*100-H*10)
150 G=INT(A-D*1000-H*100-T*10)
160 BREUK=A-D*1000-H*100-T*10-G
170 S=BREUK*20
180 IF S=>15 THEN KW=3:S=S-15
190 IF S=>10 THEN KW=2:S=S-10
200 IF S=>5 THEN KW=1:S=S-5
210 IF S=>4 THEN DB=2:S=S-4
220 IF S=>2 THEN DB=1:S=S-2
230 IF T=>5 THEN T=T-5 :V=1
240 IF G=>5THEN VJ=1:G=G-5
250 IF G=>2 AND KW=>2 THEN G=G-2:KW=KW-2
:RI=1
260 IF VJ=1 AND T=>2 THEN VJ=0:T=T-2:VT=
1
270 DU$=BR$+"duizend":HO$=BR$+"honderd
280 VI$=BR$+"vijftig":VJ$=BR$+"vijf"
290 TI$=BR$+"tien":RI$="rijksdaalder"
300 GU$="Gulden(s)":KW$="Kwartje(s)"
310 DB$="Dubbeltje(s)":S$="Stuiver"
320 VT$=BR$+"vijfentwintig
330 LOCATE 5,10:PRINTD;DU$
340 LOCATE 5,11:PRINTH;HO$
350 LOCATE 5,12:PRINTV;VI$
360 LOCATE 5,13:PRINTVT;VT$
370 LOCATE 5,14:PRINTT;TI$
380 LOCATE 5,15:PRINTVJ;VJ$
390 LOCATE 5,16:PRINTRI;RI$
400 LOCATE 5,17:PRINTG;GU$
410 LOCATE 5,18:PRINTKW;KW$
420 LOCATE 5,19:PRINTDB;DB$
430 LOCATE 5,20:PRINTS;S$
```

Schaalvergroting

Iets wat ik naar mijn idee te weinig in programma's tegenkom, is de mogelijkheid van MSX BASIC, om voorstellingen, gemaakt met het DRAW statement, te vergroten of te verkleinen.

Van deze mogelijkheid is in het volgende programma gebruik gemaakt om het cijfer 1 te laten "groeien." Dit geeft een 3D-effekt, zonder enig moeilijk programmeerwerk.

De variabele Z zorgt voor de vergroting. S is de eigenlijke SCALE-Variabele binnen de DRAW-String. Normaal staat deze variabele op 4 ingesteld.

```
1 REM ***************************
2 REM *                         *
3 REM * Een toepassing van het  *
4 REM * veel vergeten SCALE     *
5 REM * commando.               *
6 REM *                         *
7 REM ***************************
8 :
10 COLOR ,1,1
20 SCREEN2:X=140:Y=70:CO=2:Z=1
30 CO=15
40 PSET (X,Y)
50 FOR Z=1 TO  30 STEP 2
60 A$="s=z;c=co;d10 r3 d5 l10 u5 r3 u10
l5 e9 d9"
70 DRAW A$
80 NEXT Z
```

```
90 FOR T=1 TO 500: NEXT
100 CLS:GOTO 40
```

MSX effekten

SCREEN 0 is normaliter het tekstscherm, dat wordt gebruikt voor de invoer van uw programma's. Veel meer dan dat is er niet meer mogelijk vanuit de standaard BASIC opdrachten.

Met het VPOKE statement zijn de mogelijkheden onder schermmode 0 opeens sterk toegenomen.
Het volgende programma laat u dit zien.

- In de regels 30 en 40 wordt de plaats in het VRAM dat gevuld is met de code voor "!" op 0 'geVpoked'.
- In de regels 50 t/m 62 wordt het gehele scherm door middel van een machinetaal routine met de ASCII waarde van "!" geladen. Dit is gedaan in machinetaal, omdat BASIC hier eenvoudig te langzaam is.
- In de regels 170 en 180 wordt de plaats in het VRAM, dat normaal wordt ingenomen door "!" met willekeurige waarden tussen de 1 en de 255 'geVpoked'.

Omdat eerst het gehele scherm met de waarde van "!" is geladen, zal elke positie op het scherm die deze waarde heeft, veranderen als we de vorm van dit teken in het VRAM veranderen. Dit zijn alle posities op het scherm, behalve de punten die het MSX LOGO opbouwen. Deze zullen dan ook niet mee veranderen met de achtergrond.

```
1 REM ***************************
2 REM *                         *
3 REM * INTRODUKTIE SCHERM MET  *
4 REM * SPECIALE EFFECTEN...    *
5 REM *                         *
6 REM ***************************
7 :
```

```
8 CLEAR 200,&HCFFF
10 LOCATE,,0
20 ZZ=1:SCREEN0:COLOR 15,1
30 FOR X=2312 TO 2312+7
40 VPOKE X,0:NEXT
50 :
51 DATA F3,3E,21,01,C0,03,21,00,00,CD,56
,00,C9
52 :
61 T=&HD000:FOR I=0TO12:READ A$:POKE T+I
,VAL("&H"+A$):NEXT
62 DEFUSR=T:X=USR(0)
63 :
70 LOCATE 5,6: PRINT"!..!!!!..!!!.......
.!!!..!! T.M.
80 LOCATE 5,7: PRINT"!...!!...!!........
.!!!..!!
90 LOCATE 5,8: PRINT"!........!..!!!!!!!
..!..!!!
100 LOCATE 5,9: PRINT"!..!..!..!...!!!!!
!..!..!!!
110 LOCATE5,10: PRINT"!..!!!!..!!......!
!!...!!!!
120 LOCATE5,11: PRINT"!..!!!!..!!.......
!!...!!!!
130 LOCATE5,12: PRINT"!..!!!!..!!!!!!!..
!!....!!!
140 LOCATE 5,13:PRINT"!..!!!!..........!
!..!..!!!
150 LOCATE 5,14:PRINT"!..!!!!.........!!
..!!!..!!
160 GOTO 200
170 X=2312
180 BB=RND(1)*255:VPOKE X+ZZ,BB:VPOKEX-1
+ZZ,BB
190 RETURN
```

```
200 GOSUB 320
210 A=2:FOR H=1TO245
220 FORJ=1TO255STEPA:SOUND2,J:NEXTJ
230 FORJ=255TO1STEP-A:SOUND2,J:NEXTJ
240 IFS=0THENA=A+1:IFA>245THENS=2
250 IFS=2THENA=A-1
260 ZZ=ZZ+2:IF ZZ=9 THEN ZZ=1:GOSUB170 E
LSE GOSUB 170
270 NEXT H
280 PLAY"V10r","V10r","v10r"
290 J=INT(RND(1)*15+1):K=INT(RND(2)*15+1
):IFJ=KTHENJ=1:K=3
300 COLORJ,K:FORH=1TO400:NEXTH
310 GOTO 290
320 RESTORE330:FOR J=0 TO 13:READ A:SOUN
D J,A:NEXT J:RETURN
330 DATA 255,255,0,0,0,0,0,56,27,12,12,1
5,15,1
```

De vergeten paint opdracht

Bij MSX-1 computers wordt vaak geen gebruik gemaakt van de extra mogelijkheid die de paint instruktie heeft in schermmode 3.

U kunt nl. in een andere kleur inkleuren dan de buitenrand van het vak dat u wilt kleuren. U geeft dan direkt achter het paint statement de kleur waarmee u wenst in te kleuren en daarachter de kleur van de buitenrand.

Voor MSX-2 gebruikers geldt hetzelfde ook voor alle schermen vanaf SCREEN5.

```
1 REM ***************************
2 REM *                         *
3 REM *       screen 3          *
4 REM * de vergeten paint       *
5 REM *       opdracht          *
6 REM *                         *
7 REM ***************************
10 SCREEN3
20 :
30 CIRCLE(128,96),50,15
35 PAINT(128,96),3,15
40 :
50 GOTO 50
```

Sprites en karakters

Onderstaande routine maakt het mogelijk om uw gehele karakterset inklusief grafische karakters in de spriteset te kopiëren (in SCREEN1!). Dit wordt gedaan in regel 20 (VDP (6) = VDP (4)).

```
1 REM ***************************
2 REM *                         *
3 REM * De MSX sprite/karakter- *
4 REM * set....                 *
5 REM *                         *
6 REM ***************************
7 :
10 KEYOFF
20 COLOR 15,1,1:SCREEN1,1:VDP(6)=VDP(4)
30 LOCATE5,5:PRINT"Demonstratie van de M
SX
40 LOCATE5,6:PRINT"karakter set......
50 FORX=1TO255
60 FORI=1TO7.3 STEP.3
70 PUTSPRITE0,(120+SIN(I)*50,90+COS(I)*5
0),7,X
80 NEXT
90 NEXT
100 GOTO 60
```

SCREEN3 effekten

Scherm mode 3 (SCREEN3) wordt maar weinig toegepast in programma's. Dit komt waarschijnlijk omdat de resolutie te laag is en er niet direkt toepassingen zijn te vinden voor het afbeelden van grote letters.

De toepassing van deze grote letters is in de volgende routine zeer explosief. Deze routine kunt u bijvoorbeeld gebruiken bij het maken van een titelpagina bij uw eigen programma's.

```
1 ' **************************
2 ' *                        *
3 ' * DIT PROGRAMMA GEEFT EEN*
4 ' * MOOIE TOEPASSING VAN   *
5 ' * SCHERM MODE 3 VOOR HET *
6 ' * BEREIKEN VAN SPECIALE  *
7 ' * EFFECTEN MET TEKST...  *
8 ' *                        *
9 ' **************************
10 OPEN"grp:"AS1
20 X=15:Y=20:C=2
30 COLOR3,1,1:SCREEN3
40 LINE(1,1)-(255,191),15,BF
50 LINE(5,5)-(250,186),1,BF
60 :
70 :
80 '        msx gaat bewegen
90 :
100 MSX=2
110  FOR SVI = 1 TO MSX
120 X=65:Y=86:C=2
130 SVI$="MSX"
```

```
140  FOR PLAATS = 85 TO 65 STEP-5
150      FOR T=1TO3
160 COLOR C:FORI=1TOLEN(SVI$):X=X+25:PRE
SET(X,Y):PRINT#1,MID$(SVI$,I,1):NEXT
170         IFT=2THENC=C+2
180         Y=Y+5
190         X=X-PLAATS
200      NEXT
210  X=65:Y=86:C=2
220  LINE(65,60)-(190,120),1,BF
230      NEXT PLAATS
240      NEXT SVI
250 :
260 X=65:Y=86:C=2
270 SVI$="MSX"
280 COLOR C:FORI=1TOLEN(SVI$):X=X+25:PRE
SET(X,Y):PRINT#1,MID$(SVI$,I,1):NEXT
290 :
300 GOTO 100
```

INPUT met hoofdletters

De volgende routine zal u zeer veel programmeer regels schelen als u een programma schrijft waarbij u veel INPUT statements gebruikt en daarop vergelijkingen uitvoert.

Alle letters die worden ingevoerd, worden omgezet in hoofdletters, waardoor slechts een vergelijking nodig is in plaats van 2.

```
1 REM ***************************
2 REM *                         *
3 REM *   Input met alleen      *
4 REM *   hoofdletters          *
5 REM *                         *
6 REM ***************************
7 :
10 COLOR 1,14,14:SCREEN2
20 FORX=1TO255 STEP 3
30 LINE(150,X)-(X,50),1
40 NEXT
50 COLOR 1,14:SCREEN0:WIDTH40:KEY OFF
60 PRINT"Wilt u dit nog eens zien (J/N)?
"
70 :
80 *******  DE ROUTINE  *******
90 :
100 A$=INPUT$(1)
110 A$=CHR$(ASC(A$)AND &HDF)
120 IF A$="J" THEN GOTO 10 ELSE END
```

Snelheids demonstratie

Met behulp van het onderstaande programma wordt gedemonstreerd hoe u de snelheid van een BASIC programma vaak aanzienlijk op kunt voeren.
Voer het volgende programma in:

```
1 '*********************************
2 '*                               *
3 '*          SNELHEIDS            *
4 '*                               *
5 '*         DEMONSTRATIE          *
6 '*                               *
7 '*********************************
8 :
10 SCREEN 0
20 TIME=0
30 FOR A=1 TO 2000
40 NEXT A
50 PRINT TIME
```

Het programma maakt gebruik van de ingebouwde klok. Aan het begin van het programma wordt deze klok op de waarde "0" gezet (regel 20). In regel 50 wordt de inhoud van de klok op het beeldscherm "geprint".

Run dit programma en noteer de op het beeldscherm verschenen tijd.
Verander regel 40 in "NEXT" in plaats van "NEXT A" en run het programma opnieuw. Noteer de tijd.
Door het weglaten van de A in "NEXT A", is de snelheid van het programma verhoogd. De laatst genoteerde tijd was korter dan de eerst genoteerde.

Voeg nu de volgende regel toe: „9 DEFINT A." Run het programma nogmaals. Ook nu is de snelheid van het programma weer opgevoerd. Door het definiëren van de variabele A als integer, hoeft de computer minder te rekenen (een integer getal is een geheel getal, dus zonder decimalen).

Konklusie: definieer **indien mogelijk** iedere nummerieke variabele als integer. Laat achter het statement "NEXT" de variabele weg.

```
1 '********************************
2 '*                              *
3 '*   EEN UITGEWERKT VOORBEELD   *
4 '*                              *
5 '*    SNELHEIDS DEMONSTRATIE    *
6 '*                              *
7 '********************************
8 :
9 SCREEN0:CLS
10 '
11 '******************************
12 '*                            *
13 '*    ELKE INSTRUKTIE OP EEN  *
14 '*        APARTE REGEL        *
15 '*                            *
16 '******************************
17 '
18 PRINT "BEGINTIJD REGEL 19 =    0"
19 TIME=0
20 FOR A=1 TO 2000
21 NEXT A
22 PRINT "TIJD IN REGEL   22 = ";TIME
23 PRINT
24 '
25 '******************************
26 '*                            *
```

```
27 '*     ALLE INSTRUKTIES SAMEN    *
28 '*         OP EEN REGEL          *
29 '*                               *
30 '*******************************
31 '
32 PRINT "BEGINTIJD REGEL 32 =     0":TIM
E=0:FOR A=1 TO 2000:NEXT A:PRINT "TIJD I
N REGEL   32 = ";TIME
33 PRINT
34 '
35 '*******************************
36 '*                              *
37 '*  NU IS DEFINT A TOEGEVOEGD  *
38 '*                              *
39 '*******************************
40 '
41 PRINT "BEGINTIJD REGEL 42 =     0"
42 DEFINT A:TIME=0:FORA=1TO2000:NEXTA:PR
INT"TIJD IN REGEL   42 =  ";TIME
43 PRINT
44 '
45 '*******************************
46 '*                              *
47 '*      TENSLOTTE IS NU OOK    *
48 '*    DE A NA NEXT WEGGELATEN  *
49 '*                              *
50 '*      DE SNELHEID VAN HET    *
51 '*        IS NU OPTIMAAL       *
52 '*                              *
53 '*******************************
54 '
55 PRINT "BEGINTIJD REGEL 56 =     0"
56 DEFINT A:TIME=0:FORA=1TO2000:NEXT:PRI
NT"TIJD IN REGEL   56 =  ";TIME
```

Dubbele letters in SCREEN2

Soms is het wenselijk om een bepaalde tekst opvallender te maken dan andere teksten. Dit kan ondermeer gedaan worden door deze tekst dubbel te printen op het scherm. Het volgende programma bevat een routine om dit te bewerkstelligen.

Hoe dient de routine gebruikt te worden?

Deze routine werkt alleen in screenmode 2.
De werkelijke routine begint bij regel 100 en eindigt bij regel 120.
Voordat u de routine aanroept, dient u eerst een drietal variabelen een waarde te geven:

- A$: Deze variabele dient de te printen tekst te bevatten.
- X1: Deze variabele dient de X coördinaat van de te printen tekst te bevatten.
- Y: Deze variabele dient de Y coördinaat van de te printen tekst te bevatten.

Regels 10-80 bevatten twee voorbeelden die de routine aanroepen nadat aan bovenstaande parameters een waarde is toegekend.

```
1 '   *****************************
2 '   *                           *
3 '   *   DUBBEL  DIKKE LETTERS   *
4 '   *                           *
5 '   *      IN SCREEN 2          *
6 '   *                           *
7 '   *****************************
```

```
8 :
10 SCREEN2
20 OPEN "grp:" AS 1
30 A$="STARK TEXEL"
40 X1=50:Y=30
50 GOSUB 100
60 A$="TRUUKS & TIPS 3"
70 X1=50:Y=40
80 GOSUB 100
90 GOTO 90
100 FOR X=X1 TO X1+1 STEP.5
110 PSET(X,Y),PEEK(&HF3EA):PRINT#1,A$
120 NEXT:RETURN
```

Sprite botsing routine

Met MSX BASIC is het mogelijk om een subroutine aan te laten roepen als twee sprites tegen elkaar botsen. Dit gebeurt met "ON SPRITE GOSUB regelnummer" en "SPRITE ON". Bij een botsing tussen twee sprites wordt er naar een subroutine gesprongen.
Het is echter niet mogelijk om op te vragen tegen welk sprite nummer een bekende sprite botste (b.v. een kogel). Dit kan b.v. nodig zijn in een spel als "space invaders". Als een bekende sprite, in dit geval een kogel, een ruimteschip raakt, in dit geval de onbekende sprite, dan moet dit ruimteschip verdwijnen daar het vernietigd is door de kogel. Er zijn echter meerdere ruimteschepen. Het is niet bekend welk ruimteschip de kogel raakte, het is wel bekend dat de kogel een ruimteschip raakte (via de "on sprite gosub" routine).
Onderstaand programma bevat een routine die uitzoekt tegen welk sprite nummer de bekende sprite botste.
Als bekend sprite nummer dient altijd sprite 0 genomen te worden. De onbekende sprite nummers kunnen variëren van 1-15. De eigenlijke routine loopt van regel 190-230.
Het programma demonstreert ook het gebruik van de routine. Beweeg de bekende sprite met de cursor besturing. Als de bekende sprite botst tegen een andere, dan verschijnt in beeld tegen welk nummer de bekende sprite botste.

```
1 '*********************************
2 '*                               *
3 '* DIT PROGRAMMA STELT U IN STAAT*
4 '* OM NA TE GAAN TEGEN WELKE     *
5 '* SPRITE, SPRITEO BOTSTE.       *
6 '*                               *
7 '*********************************
8 :
```

```
10 SCREEN1:KEYOFF:DEFINTA-Z
20 VDP(6)=VDP(4)
30 ONSPRITE GOSUB 190
40 FOR U=1 TO 15:X=X+15:Y=Y+10
50 PUTSPRITE U,(X,Y),15,U:NEXT:'de onbek
ende sprites (1-15)
60 M=5:N=5:SPRITEON
70 PUTSPRITE0,(M,N),15,1 :'de bekende sp
rite (0)
80 U=STICK(0)
90 IFU=1ANDN=>0 THEN N=N-1:GOTO70
100 IFU=5ANDN<=184 THEN N=N+1:GOTO70
110 IFU=3ANDM<=252 THEN M=M+1:GOTO70
120 IFU=7ANDM>=0 THEN M=M-1:GOTO70
130 GOTO70
140 :
150 REM * routine die uitzoekt tegen *
160 REM * welk sprite nummer sprite0 *
170 REM * botst. regels 190-230      *
180 :
190 SPRITEOFF:FORP=4TO60STEP4:Q=P+6912
200 R=VPEEK(Q):IFR=209THENNEXTELSEVPOKEQ
,209:FORS=1TO50:NEXT
210 T=(VDP(8)AND32):IF T=0 THEN PRINT"sp
rite";P/4;"botste met sprite 0":RETURN60
220 VPOKE Q,R:NEXT
230 RETURN60
```

Speciale toetsen uitlezen

Heeft u zich ook al eens afgevraagd waarom uw computer is voorzien van een SELECT-toets?
Deze toets wordt normaal niet ondersteund door MSXBASIC. Slechts enkele programma's gebruiken deze toets om bijvoorbeeld een menu aan te roepen.

Van nu af aan kunt ook u gebruik maken van deze toets en de vele andere, normaal niet uit te lezen, toetsen (zoals CAPS LOCK, INSERT, DEL, CODE, TAB, BACKSPACE).

Het programma maakt gebruik van de DEF FN-instruktie, maar wel in een vorm die u zeer weinig tegenkomt. De funktie G(B) (in regel 10) geeft de waarde −1 indien de funktie waar is, in het andere geval een 0.

Het lijkt alsof in regel 110 een IF-THEN konstruktie staat, waarbij geen vergelijking voorkomt. Het tegendeel is echter waar. Alleen als Z 〈 〉 0, wordt er naar regel 300 gesprongen. In het andere geval loopt het programma gewoon door.

Ter illustratie staan in regels 30-90 en 270-300 enkele voorbeelden gegeven, die betrekking hebben op het uitlezen van de in de listing genoemde toetsen.

```
1 '********************************
2 '*                              *
3 '* MET DEZE ROUTINE KUNT U      *
4 '* NAGAAN OF EEN VAN DE SPECI-  *
5 '* ALE TOETSEN, B.V. DE SELECT  *
6 '* TOETS, INGEDRUKT WORDT.      *
7 '********************************
8 :
```

```
10 DEFFN G(B)=(INP(&HA9)ANDB)=0
20 :
30 A=7:B=64:GOSUB 310 :'SELECT TOETS
40 IF Z THEN GOTO 270
50 A=6:B=8:GOSUB 310  :'CAPS LOCK
60 IF Z THEN GOTO 280
70 A=7:B=16:GOSUB 310 :'STOP TOETS
80 IF Z THEN GOTO 290
90 A=8:B=4:GOSUB 310  :'INS TOETS
100 :
110 IF Z THEN GOTO 300
120 :
130 ' A=8:B=8                  DEL TOETS
140 ' A=6:B=16                CODE TOETS
150 ' A=6:B=4                GRAPH TOETS
160 ' A=7:B=32           BACKSPACE TOETS
170 ' A=7:B=8                  TAB TOETS
180 ' A=7:B=4                  ESC TOETS
190 ' A=8:B=2                  CLS TOETS
200 ' A=6:B=32                F1   TOETS
210 ' A=6:B=64,B=128       F2,F3   TOETS
220 ' A=6:B=1                SHIFT TOETS
230 ' A=6:B=2                 CTRL TOETS
240 ' A=7:B=128    ENTER/RETURN TOETS
250 GOTO 20
260 :
270 PRINT"U heeft de select toest ingedr
ukt":GOTO 30
280 PRINT"U heeft op de capslock toets g
edrukt":GOTO30
290 PRINT"U heeft op de stop toets gedru
kt":GOTO 30
300 PRINT"U heeft op de ins toets gedruk
t":GOTO 30
310 OUT&HAA,((INP(&HAA)AND&HF0)OR A): Z=
FN G(B):RETURN
```

Lange funktietoetsen

MSX computers beschikken over 10 programmeerbare funktietoetsen. Deze toetsen kunnen per stuk maximaal 15 karakters bevatten. Meestal is dit wel voldoende. Echter, in enkele gevallen is het deze lengte niet toereikend.

Onderstaande routine geeft u de mogelijkheid om in funktietoetsen 1-7 een tekst te plaatsen van ruim 30 karakters.
De tekst van de funktietoets dient u in A$ te plaatsen (regel 20).

In dit geval bevat de A$ de tekst:

SCREEN2:FORX=1T099:CIRCLE (99,96),X:NEXT

Probeert u deze programma regel maar eens in een funktietoets te plaatsen door middel van de KEY-instruktie.

Opmerking: Indien een funktietoets meer dan 15 tekens bevat, dan zal een deel van de tekst ook in andere funktietoetsen terecht komen. De tekst is nu echter wel door middel van een druk op een enkele funktietoets op te roepen.

```
1 '*********************************
2 '*                               *
3 '* DEZE ROUTINE STELT U IN STAAT *
4 '* OM EEN FUNCTIE TOETS MEER DAN *
5 '* 15 TEKENS TE LATEN BEVATTEN   *
6 '* (P.S. ALLEEN F1 T/M F7)       *
7 '*********************************
8 :
10 SCREEN0:WIDTH40
20 A$="screen2:forx=1to99:circle(99,96),
```

```
x:NEXT"
30 INPUT"In welke functie toets wilt u d
e waarde van a$ zetten ( 1 t/m  7)";A
40 IF A>7 OR A<1THEN PRINT:PRINT"Alleen
de functie toetsen 1 t/m 7 kunnen verand
erd worden":PRINT:GOTO30
50 T=&HF87F+(A-1)*&H10
60 B=LEN(A$)
70 FOR Z=1 TO B:M$=MID$(A$,Z,1):N=ASC(M$
):POKE T,N:T=T+1:NEXT:POKE T,0
80 CLS
```

Funktietoetsen bewaren

Als u uw funktietoetsen heeft veranderd, dan zijn deze verdwenen, als u uw computer uitzet. U kunt ze echter voor verlies veilig stellen als u het gebied van het geheugen waar de funktietoetsen liggen opgeslagen wegschrijft (DISK/TAPE).

De volgende routine laat u dit alles duidelijk zien. Aan u de taak om het nuttig in uw programma's te verwerken.

```
1   REM ***************************
2   REM *                         *
3   REM * FUNCTIE TOETSEN STAAN   *
4   REM * IN HET GEHEUGEN         *
5   REM *                         *
6   REM * VANAF &HF87F TOT &HF91E *
7   REM *                         *
8   REM ***************************
9 :
10 COLOR 1,14
20 SCREEN0:WIDTH38:KEYON
30 :
40 KEY 1," MSX
50 KEY 2,"TRUUKS
60 KEY 3,"  EN
70 KEY 4,"TIPS
80 KEY 5,"DEEL 3
90 PRINT"FUNCTIETOETSEN WEGSCHRIJVEN (J/
N)"
100 A$=INPUT$(1)
110 A$=CHR$(ASC(A$)AND223)
120 IF A$="J" THEN BSAVE"TOETSEN.OBJ",&H
```

```
F87F,&HF91E
130 DEFUSR=&H3E:X=USR(0) :'FUNCTIE TOETS
EN WEER NORMAAL.
140 SCREEN0
150 PRINT"FUNCTIE TOETSEN LADEN (J/N)"
160 A$=INPUT$(1)
170 A$=CHR$(ASC(A$)AND223)
180 IF A$="J" THEN BLOAD"TOETSEN.OBJ"
190 SCREEN0
200 END
```

DOS vanuit BASIC

Het opstarten van een BASIC programma vanuit MSX DOS is heel eenvoudig.

nl.: A ⟩ BASIC filenaam (enter)

BASIC wordt dan opgestart en het programma (filenaam) gerund.

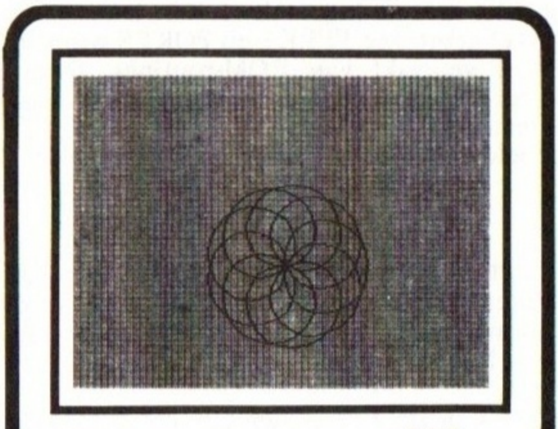
Het opstarten van een programma onder MSX DOS vanuit DISK BASIC is op het eerste gezicht echter niet zo eenvoudig.

Onderstaande routine laat zien hoe u door middel van een druk op funktietoets nr.10 een programma onder MSX DOS kunt opstarten.

"filenaam" mag elk MSX DOS programma zijn met de toevoeging ".COM".

```
1 REM ***************************
2 REM *                         *
3 REM *    DOS VANUIT BASIC     *
4 REM *                         *
5 REM ***************************
6 :
10 KEY10,"_system"+CHR$(13)+"filenaam"+C
HR$(13)
```

PEEKS, POKES en ROM-routines



In veel programma's worden de statements PEEK, POKE, VPEEK en VPOKE gebruikt. Het gebruik hiervan is slechts zinvol als de programmeur goed op de hoogte is van de geheugen indeling van zijn/haar MSX. Zo kunt u met het PEEK kommando elk geheugenadres "uitlezen". Met POKE kunt u elk geheugenadres "beschrijven". VPEEK en VPOKE hebben ook deze funktie, zij het nu voor het Video geheugen, dat zorgt voor de schermopbouw.
Een goed voorbeeld van een handige toepassing van de POKE opdracht is het programma dat ervoor zorgt dat een "geCSAVED" programma automatisch wordt gerund. Hier wordt het woord "RUN" in de keyboard buffer gePOKED. Zodra de MSX weer terug komt met OK, dan wordt de "RUN" instruktie uitgevoerd met een automatische carriage return.
Het gebruik van het VPOKE statement kunt u zeer duidelijk terugvinden in het "MSX effekten" programma. Voor nadere bijzonderheden omtrent het gebruik van deze instruktie verwijzen wij u naar de begeleidende tekst.
Naast de veel gebruikte PEEKS en POKES wordt ook nogal eens gebruik gemaakt van ROM-routines. Deze routines worden aangeroepen vanuit BASIC door middel van DEFUSR = ROMadres:X=USR (0).
Een voorbeeld van deze instruktie is bijvoorbeeld:

DEFUSR=&h000:X=USR (0)

Dit resulteert in een warme start, oftewel een RESET van uw computer. In de programma's "Sprites verdwijnen", "Cursor spring in het veld" en andere wordt ook gebruik gemaakt van ROM-routines.

Om u zelf ook in staat te stellen om wat te experimenteren met wat PEEKS en POKES volgt hier een overzicht van enkele belangrijke systeem variabelen. Deze variabelen wor-

den door uw computer gebruikt om alle zaken op een rijtje te zetten. De systeemvariabelen bevinden zich boven in het RAM geheugen.

Om een adres uit te lezen gebruik:

Z=PEEK (LAGE BYTE) + 256 * PEEK (HOGE BYTE)

LAGE BYTE	HOGE BYTE	FUNKTIE
&HF676	&HF677	START VAN EEN BASIC PROGRAMMA
&HFC4A	&HFC4B	TOP VAN HET BASIC GEHEUGEN
&HFCA0	&HFCA1	HUIDIGE INTERVAL WAARDE

De nu volgende adressen bestaan slechts uit 1 byte, welke kunnen worden uitgelezen met:

PEEK (ADRES)

ADRES	FUNKTIE
&HF3B0	REGEL BREEDTE
&HF6AA	AUTO REGELNUMMERING VLAG (1=AAN / 0=UIT)
&HF85F	AANTAL FILES – MAXFILES
&HF3DB	CLICK AAN/UIT (1=AAN / 0= UIT)
&HF3DC	CURSOR REGEL
&HF3DD	CURSOR COLOM
&HF3E9	VOORGROND KLEUR
&HF3EA	ACHTERGROND KLEUR
&HF3EB	BORDER KLEUR
&HFCAB	HOOFDLETTER (JA/NEE) (1=AAN / 0=UIT)

&HFCAF SCHERM MODE NUMMER

Voordat u zelf aan de slag gaat met deze systeemvariabelen, volgen hier nog enkele voorbeelden voor het gebruik.

POKE &HFCAB,1 ==»	Alle letters worden als hoofdletters op het scherm gebracht.
POKE &HFCAB,0 ==»	Alle letters worden als kleine letters op het scherm gebracht.
PRINT PEEK (&HF3E9) ==»	Geeft de op dat moment geaktiveerde voorgrond kleur.

PRINT PEEK (&HF676) + 256 * PEEK (&HF677)
Geeft het startadres van een BASIC programma.

Veel plezier met het gebruik van voorgenoemde adressen. Experimenteert u rustig met de PEEK en POKE instrukties. Uw computer kan er niet defekt door raken.

Laden en runnen met CLOAD

Het automatisch laten "RUNNEN" van een met CSAVE weggeschreven programma is niet zomaar mogelijk. Alleen een als ASCII-file weggeschreven programma kan direkt worden gerund na het laden. Voor het CLOAD probleem is de oplossing gegeven in de vorm van een kleine routine.

Deze routine kunt u toepassen bij uw eigen programma's op tape die elkaar moeten opstarten.

De routine dient u als ASCII-file weg te schrijven (SAVE "CAS: filenaam") voor het eigenlijke programma (dat met CSAVE wordt weggeschreven).

```
1 REM ****************************
2 REM *                          *
3 REM * CLOAD + RUN.....         *
4 REM *                          *
5 REM ****************************
6 :
10 KEYOFF:COLOR15,4,4:SCREEN 0:LOCATE 10
,10:PRINT"LOADING...."
20    Q$="RUN"+CHR$(13):FORQ=-1040TO-1000
:POKEQ,0:NEXT:POKE-3078,240:POKE-3077,25
1:POKE-3076,240+LEN(Q$):POKE-3075,251:FO
RQ=1TOLEN(Q$):POKEQ-1041,ASC(MID$(Q$,Q,1
)):NEXT:CLOAD
```

DISKBASIC → MSXDOS

Diskgebruikers, welke in het bezit zijn van het operating system MSX DOS zullen vast weten dat het mogelijk is om vanuit DISK BASIC naar MSX DOS over te schakelen. Dit gebeurd door middel van het kommando CALL SYSTEM.

Dit is gewoonlijk slechts mogelijk als u met MSX DOS bent opgestart.

Bent u niet in MSX DOS opgestart dan is het normaal niet mogelijk om toch door middel van het CALL SYSTEM kommando in MSX DOS te geraken.

De volgende routine maakt dit echter wel mogelijk.

```
1 REM ****************************
2 REM * OP ADRES &HF346          *
3 REM * KIJKT UW COMPUTER        *
4 REM * OF ER MET MSXDOS IS      *
5 REM * OPGESTART (J/N)          *
6 REM ****************************
7 :
10 POKE &HF346,0    :'GEEN DOS
20 POKE &HF346,1    :'WEL  DOS
30 :
40 REM ALS U DUS BENT OPGESTART
50 REM IN DISK BASIC EN U WILT
60 REM GEBRUIK MAKEN VAN MSXDOS
70 REM DAN KUNT U NA POKE&HF346,1
80 REM MET  SYSTEM IN DOS KOMEN.
90 :
```

Sprites verdwijnen

Met de nu volgende routine is het mogelijk om in een keer alle sprites van het scherm te laten verdwijnen.

Deze routine bevindt zich in het ROM geheugen van uw MSX op locatie &H6A8.

In regel 50 wordt bovendien een truukje uitgehaald, waardoor alle karakters in de sprite tabel worden gekopieerd (VDP (6) = VDP (4)).

```
1 REM **************************
2 REM *                        *
3 REM * Het in een keer wissen *
4 REM * van alle sprites ==>>  *
5 REM *                        *
6 REM * DEFUSR=&H6A8           *
7 REM *                        *
8 REM **************************
9 :
10 STOP ON :ON STOP GOSUB 190
20 Z=1
30 KEY OFF
40 SCREEN1
50 VDP(6)=VDP(4)
60 :
70 FOR T=1TO 5
80 READ A:READ B
90 FOR X=80 TO 140 STEP 20
100 FOR Y=ATO B
```

```
110 PUT SPRITE Z,(X,Y),15
120 NEXT:Z=Z+1:IF Z= 31 THEN GOTO 200
130 NEXT
140 NEXT
150 LOCATE 2,23:PRINT"Clear all sprites.
..
160 FOR T=1TO 200:NEXT
170 DEFUSR=&H6A8:X=USR(0) :'clear sprite
s
180 GOTO 180
190 SCREEN0:WIDTH38:LIST
200 GOTO 200
210 DATA 1,30,30,60,60,90,90,120,120,150
```

Cursor-spring in het veld

Het volgende programma is slechts een aardigheidje om te laten zien wat je zoal kunt doen met het gebruik van ROM-routines.

In dit programma gebruiken we de routine voor CTRL-B (Cursor naar het vorige woord) en de routine voor CTRL-F (Cursor naar het volgende woord).

```
1 REM *******************************
2 REM *                             *
3 REM * DEFUSR=&H260E  -  CTRL-B    *
4 REM *                             *
5 REM * DEFUSR=&H25F8  -  CTRL-F    *
6 REM *                             *
7 REM *******************************
8 :
10 SCREEN0:WIDTH40
20 LOCATE 0,5:PRINT"THE MSX CURSOR IN AC
TION ON THE 40*24":PRINT"TEXT SCREEN...
30 LOCATE 39,6,1
40 ON ERROR GOTO 120
50 DEFUSR=&H260E
60 FORT=1TO500:NEXT
70 X=USR(0)
80 IF CSRLIN=0 THEN GOSUB 140
90 FOR T=1TO 200:NEXT
100 GOTO 70
110 GOTO 110
120 RESUME 130
130 END
140 :
```

```
150 DEFUSR2=&H25F8
160 FORT=1TO300:NEXT
170 X=USR2(0)
180 IF CSRLIN>=22 THEN RETURN
190 GOTO 160
```

Scherm aan/uit

De volgende routines zorgen ervoor dat het scherm aan/uit wordt gezet. Daarbij wordt het grafische scherm eens niet geïnitialiseerd door het SCREEN2 kommando, maar door het aanroepen van een ROM-routine. Op deze manier zou dat ook gedaan zijn in een machinetaal programma.

Deze routine is zeer goed in een van uw eigen programma's toe te passen. U kunt bijvoorbeeld een beeld opbouwen zonder dat men het ziet. Pas als het beeld geheel is opgebouwd zet u het scherm aan. (vgl. het voorbeeld)

```
1 REM ***************************
2 REM *                         *
3 REM * DEFUSR=&H5D2 - SCREEN2  *
4 REM * DEFUSR=&H577 - DISABLE  *
5 REM *                SCREEN   *
6 REM * DEFUSR=&H570 - ENABLE   *
7 REM *                SCREEN   *
8 REM ***************************
9 :
10 DEFUSR=&H5D2:X=USR(0) :'screen2
20 DEFUSR=&H577:X=USR(0) :'disable
30 CIRCLE(128,96),50,15
40 PAINT(128,96),15
50 DEFUSR=&H570:X=USR(0) :'enable
60 GOTO 60
```

Het kleine lettervretertje

Het nu volgende programma demonstreert het gebruik van een ROM-routine in een BASIC programma. De hier gebruikte routine (&H2550) zorgt ervoor dat als u op de DELete toets drukt, het karakter onder de cursor verdwijnt.

Allereerst wordt in regel 10 A$ gedefinieerd.

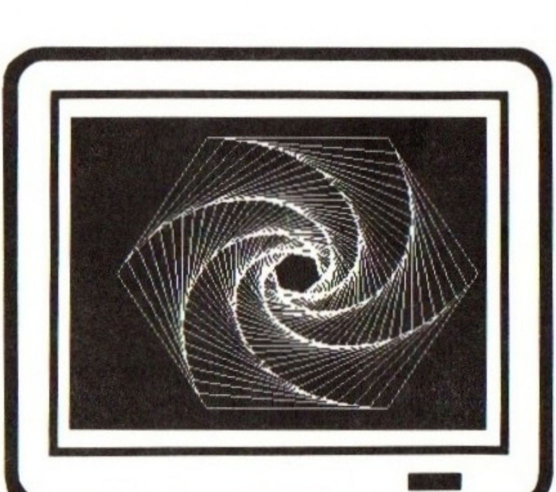
Dan wordt in de regels 20-60 de string op het scherm geplaatst met steeds 2 spaties er tussen.

In de regels 80-140 wordt willekeurig op het scherm de DELete aktie uitgevoerd. Hierdoor zal het effekt ontstaan, dat het gehele scherm langzaam maar zeker vanaf de rechterkant wordt opgegeten.

```
1 REM ***************************
2 REM *                         *
3 REM * DEFUSR=&H2550 = DELETE  *
4 REM *                         *
5 REM ***************************
6 :
10 COLOR 15,1:SCREEN0:WIDTH40:A$="Stark-
Texel"
20 Y=1
30 FOR X=0 TO (40-LEN(A$)) STEP LEN(A$)+
2:LOCATE X,Y:PRINTA$
40 IF X>=36-LEN(A$) THEN Y=Y+1:GOTO 30
50 IF Y=22 THEN 80
60 NEXT
70 :
80 FOR XX=1 TO 500
```

```
90 LOCATE RND(1)*38,RND(1)*22
100 :
110 DEFUSR=&H2550
120 X=USR(0)
130 :
140 NEXT
150 END
```

MSX kunstwerken



MSX computers behoren op grafisch gebied tot de betere in homecomputer land. Niet in de laatste plaats door de vele BASIC kommando's die het mogelijk maken om op zeer eenvoudige wijze gebruik te maken van deze kapaciteit. Het blijft echter wel een kunst om met behulp van deze krachtige kommando's een goed programma te schrijven.
Om u een idee te geven van wat er zoal mogelijk is, hebben wij enige grafische toepassingen voor u ontwikkeld, welke de standaard kommando's verenigen met wat wiskunde. Hierdoor ontstaan de mooiste voorstellingen. Is uw nieuwsgierigheid geprikkeld, probeer dan snel de programma's in dit hoofdstukje uit.

MSX kunstwerk

Ook dit programmaatje hoort weer thuis in de kategorie "truuks met graphics".

Dit soort programmaatjes zijn vaak erg in trek, maar zijn echter zeer op zichzelf staand.

U kunt er uw computer mee demonstreren aan nieuwsgierige vrienden. Deze zullen zeker onder de indruk raken.

```
10 REM ***********************
20 REM *                     *
30 REM * DRAAIENDE RECHTHOEK *
40 REM *                     *
50 REM ***********************
60 :
70 DEFSNG A-Z
80 COLOR 1,14,4:SCREEN 2
90 :
100 FOR Z=1 TO 8
110 REM *** BUITEN LUS ***
120 R=R+4*Z:N=N+R
130 X=R:Y=0:X1=0:Y1=R
140 P=Z/N:N=N/4
150 :
160 FOR I=0 TO N
170 REM *** BINNEN LUS ***
180 X=X-Y*P:Y=Y+X*P
190 X1=X1-Y1*P:Y1=Y1+X1*P
200 :
210 LINE((128+X),96+Y)-((128+X1),96+Y1)
220 LINE((128-X),96-Y)-((128-X1),96-Y1)
```

```
230 LINE((128+X1),96+Y1)-((128-X),96-Y)
240 LINE((128-X1),96-Y1)-((128+X),96+Y)
250 :
260 NEXT I                                :
REM *** BINNEN LUS EINDE ***
270 :
280 NEXT Z: REM *** BUITEN LUS EINDE ***
290 :
300 GOTO 300: REM *** EINDE ***
```

Graphic generator

Het volgende programma stelt u in staat om via uw computer allerlei verschillende voorstellingen te laten tekenen na het opgeven van enkele waarden.

Voorbeelden staan in het programma vermeld in de regels 40-90.

Experimenteer met wat andere getallen en sta telkens weer versteld van het resultaat.

```
1 REM ***************************
2 REM *                         *
3 REM * Graphic designer        *
4 REM *                         *
5 REM ***************************
6 :
10 PI=3.14159
20 RD=2*PI
30 S=PI/100
40 INPUT"K1";K1:' Bijv. 25 - Bijv. 50
50 INPUT"K2";K2:' Bijv. 25 - Bijv. 25
60 INPUT"K3";K3:' Bijv. 10 - Bijv. 10
70 INPUT"K4";K4:' Bijv. 25 - Bijv. 20
80 INPUT"K5";K5:' Bijv. 25 - Bijv. 10
90 INPUT"K6";K6:' Bijv. 10 - Bijv.  5
100 X0=125:Y0=75
110 SCREEN2:COLOR1,11,6
120 PSET(X0,Y0),6
130 XA=X0:YA=Y0
140 FORX=0TORDSTEPS
150 :   XP=X0+K1*COS(X)-K2*COS(K3*X)
```

```
160 :   YP=Y0+K4*SIN(X)-K5*SIN(K6*X)
170 LINE(XA,YA)-(XP,YP),6
180 XA=XP:YA=YP
190 NEXTX
200 FORI=0TO50
210 BEEP
220 NEXTI
230 GOTO230
```

Hexagon

De grafische toepassingen van MSX BASIC zijn zeer veelzijdig. Er worden dan ook zonder veel moeite met behulp van enige wiskunde de mooiste voorstellingen op het scherm getoverd.

Onderstaand programma demonstreert op een zeer overtuigende manier het gebruik van een kombinatie van goniometrische funkties en MSX-kunst.

```
1 REM ******************************
2 REM *                            *
3 REM * Het tekenen van een hexagon *
4 REM *                            *
5 REM ******************************
6 :
10 PI=3.14159
20 C=COS(PI/3):S=SIN(PI/3)
30 C1=COS(PI/36):S1=SIN(PI/36):SF=.95
40 X=95:Y=0:CX=140:CY=96:SC=1.16
50 COLOR15,1,1:SCREEN2
60 FORJ=1TO40
70 FORI=0TO6
80 SX=X*SC+CX:SY=CY+Y
90 IFI=0THEN PSET (SX,SY),15
100 LINE -(SX,SY),15
110 XN=X*C-Y*S:Y=X*S+Y*C:X=XN
120 NEXT I
130 XN=SF*(X*C1-Y*S1):Y=SF*(X*S1+Y*C1):X
=XN
140 NEXT J
150  GOTO150
```

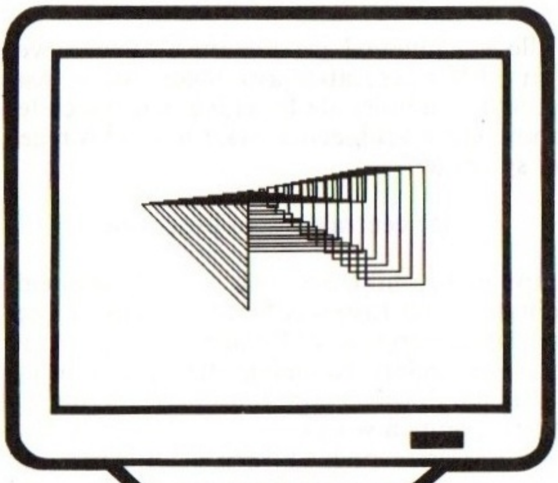
Bloem

Net zoals het hexagon is deze bloem het produkt van de "LINE" opdracht in uw MSX en wat wiskunde. Het resultaat mag er ook dit keer zijn. U ziet dat u al de verkregen wiskundekennis ook kunt aanwenden om mooie plaatjes te maken op uw computer. Waar wiskunde toch al niet goed voor is...

```
1 REM ******************************
2 REM *                            *
3 REM *  Een mooie bloem op uw MSX  *
4 REM *                            *
5 REM ******************************
6 :
10 COLOR 15,1,1:SCREEN2:SC=1.16
20 CX=140:CY=96
30 C=COS(.1):S=SIN(.1)
40 :
50 FOR TH =0 TO 6.3 STEP.05
60 R=88*COS(2*TH)
70 COLOR4:IF R<0 THEN COLOR8
80 X=CX+SC*R*COS(TH)
90 Y=CY+R*SIN(TH)
100 LINE(140,96)-(X,Y)
110 NEXT TH
120 :
130 COLOR 5
140 FOR R=90 TO 95 STEP.8:X=R:Y=0
150 FOR I=0TO63
160 T=X*C-Y*S
170 Y=X*S+Y*C:X=T
180 IF I=0 THEN PSET(CX+SC*X,CY-Y)
```

```
190 LINE-(CX+SC*X,CY-Y)
200 NEXT I,R
210 GOTO 210
```

Korte inleiding tot machinetaal



Deze korte inleiding in machinetaal heeft uitsluitend voor ogen om u enigzins vertrouwd te maken met deze vaak door mysterie omringde taal. Meestal, als er over machinetaal wordt gesproken, denken de meeste mensen dat het gaat om iets dat alleen voorbehouden is aan een selekt aantal freaks. Niets is echter minder waar! Machinetaal is aanzienlijk krachtiger dan BASIC, maar is daarentegen opgebouwd uit zeer veel korte kommando's. Om te doen wat in BASIC in een enkele opdracht gedaan kan worden, zijn in machinetaal vaak tientallen opdrachten noodzakelijk. Dit maakt het programmeren in deze taal nogal onoverzichtelijk. Moeilijk is het echter niet. De in dit boek beschreven routines demonstreren duidelijk, dat in machinetaal meer mogelijk is. Het is dan ook duidelijk, dat in machinetaal meer mogelijk is. Het is dan ook bij uitstek een taal om alles uit uw MSX te halen wat er in zit. Als u echt interesse heeft om de taal te leren, dan heeft u aan het MSX machinetaal handboek een goede handleiding.

Vrijwel alle machinetaal programma's hebben we voorzien van een in REM's geplaatste assembler listing. Assembler is een taal, die in symbolen de feitelijke machinecode aangeeft. De bijbehorende machinecode staat in DATA regels voor de assembler symbolen.

De centrale micro processor

Elke computer beschikt over een centrale eenheid die voor de coördinatie zorgt tussen computer, mens en gegevens. Bij MSX computers zorgt de Z80 hiervoor. Deze processor bemoeit zich met iedere handeling die de computer verricht. De gegevens die de processor aangeboden krijgt bepalen de aktie die ondernomen wordt.
Omdat een processor nauwelijks in staat is om gegevens zelf te bewaren, worden ze tijdelijk of permanent opgeslagen in

een geheugen. Als ze tijdelijk opgeslagen worden dan spreekt men van RAM in het andere geval van ROM. De gegevens die opgeslagen zijn/worden in het RAM kunnen veranderd worden, zij die opgeslagen liggen in het ROM kunnen niet veranderd worden.

De opbouw van de Z80

De nu volgende tekst beschrijft in het kort de opbouw van de Z80.
De centrale eenheid van de Z80 is de ALU. Deze ALU zorgt voor alle logische en arimetische bewerkingen. ALU is dus eigenlijk de centrale rekeneenheid van de Z80. Naast de ALU is de Z80 opgebouwd uit een aantal registers. Dit zijn geheugens die zich in de processor zelf bevinden. Deze registers kunnen de inhoud van iedere geheugenplaats (RAM, ROM) bevatten.
Verder beschikt de Z80 over een databus (8 bits). Via deze bus (leiding) kunnen gegevens van/naar geheugen en randapparatuur ontvangen/gezonden worden. Als bijvoorbeeld een register met een gegeven (inhoud) van een bepaalde geheugenplaats geladen worden, dan gebeurt dat via de databus. Zendt de Z80 een letter naar het beeldscherm, dan gebeurt dit wederom via de databus.

Als u in machinetaal programmeert, dan bestuurt u in feite zelf het geheel wat hierboven is beschreven. U geeft direkt de instrukties aan de Z80. Er zijn er echter maar een beperkt aantal. Het is dus noodzakelijk te weten welke instrukties de Z80 kent.
Iedere instruktie bestaat uit een getal of een reeks van getallen. De instrukties tezamen vormen de machinetaal. Machinetaal bestaat dus alleen uit getallen. Daar het vrijwel niet te doen is om deze getallen uit het hoofd te leren, heeft men woorden bedacht die een instruktie voorstellen (assemblertaal). Met behulp van een speciaal programma worden deze woorden omgezet in getallen (machinetaal). Zo'n programma wordt een assembler genoemd.
Hieronder volgen enkele voorbeelden van de assembler in-

strukties.

1 LD A,10
2 LD A,100
3 ADD A,10
4 SUBC A,10

Uitleg: Zoals reeds verteld is de processor ondermeer opgebouwd uit registers. Een van deze registers wordt het A register genoemd.
Het eerste voorbeeld betekent: "laad het A register met de waarde 10", "LD" staat voor LOAD of wel in het nederlands "LAAD". Analoog de betekenis van het eerste voorbeeld, staat voorbeeld 2 voor: "laad het A register met de waarde 100." Het derde voorbeeld betekent dat de waarde 10 bij het A register opgeteld moet worden. Tenslotte voorbeeld 4. Deze instruktie vermindert het A register met 10.

ROM routines

ROM routines zijn kleine machinetaal programma's die zich in het ROM bevinden. Er wordt tijdens het programmeren in machinetaal van deze routines vaak gebruik gemaakt omdat ze het programma sterk vereenvoudigen. Het zijn als het ware kant en klare subroutines.
Er is een ROM routine die een karakter op het beeldscherm print, die iets berekent, die kijkt of de CTRL en STOP toets ingedrukt zijn enz. enz.

Tot zover deze korte inleiding in machinetaal. We hopen dat uw interesse in deze taal toegenomen is. Als u niet alles begrepen heeft, maakt u zich dan geen zorgen. Voor de programma's in dit boek is beslist geen kennis van machinetaal vereist.

Geheugen uitbreiding onder BASIC

P.S. Deze routine is alleen bruikbaar op MSX-1 computers met een minimum geheugen van 64K.
De routine werkt voor zover wij weten op elke 64K MSX machine, behalve op de Philips VG 8030 en de SONY HB-F500P.

Het programma stelt u in staat om over een BASIC geheugen te beschikken van 2 keer 24K indien uw computer uitgerust is met 64K RAM.

Normaal, als u onder MSX-BASIC werkt, kunt u over slechts 32K RAM geheugen beschikken. Daarvan blijven er netto 28815 bytes over om te gebruiken voor uw BASIC programma's. Dit is het gevolg van de beperking van de bij MSX toegepaste microprocessor (de Z80A).
Deze processor is door zijn 8 bits struktuur slechts in staat om 64K geheugen gelijktijdig te adresseren. Door middel van een techniek die bankswitching wordt genoemd is een MSX computer echter in staat om tot een maximum van 1024K geheugen te adresseren. Dit geheugen is dan echter alleen beschikbaar voor de gevorderde machinetaal programmeur. De BASIC programmeur blijft helaas opgescheept met de 28815 bytes. Bij de MSX-2 range is dit "probleem" opgelost door middel van een RAM-disk. Dit is een soort diskdrive, maar dan in het geheugen.

Voor de MSX-1 gebruiker wordt met onderstaande routine de mogelijkheid geschapen, om twee BASIC-programma tot max. 24K in het geheugen te hebben. U heeft voor het gebruik van deze mogelijkheid de beschikking over een nieuwe instruktie, nl. "CMD". Deze instruktie wisselt de inhoud van de twee geheugen banken met elkaar om.

U kunt dit kommando (CMD) altijd gebruiken in direkt mode.
In een programma kunt u het ook gebruiken. U dient zich echter te realiseren, dat het programma wordt beëindigd nadat het CMD kommando gegeven is. Na het CMD kommando staat het BASIC programma, dat in het ongebruikte geheugendeel stond, op de plaats van het programma waarin de CMD instruktie werd aangeroepen.

Na het CMD kommando wordt altijd schermmode 0 (screen0) geïnitialiseerd.

```
1 REM ***************************
2 REM *                         *
3 REM *    SOFTWAREMATIGE       *
4 REM *    GEHEUGEN UITBREIDING *
5 REM *    D.M.V. EEN EXTRA     *
6 REM *    COMMANDO (CMD)       *
7 REM *                         *
8 REM ***************************
9 :
10 REM * KIJKEN IN WELKE BANK *
20 REM * ZICH DE 64K RAM BE-  *
30 REM * VINDT                *
40 REM *                      *
50 :
60 CLEAR 200,&HE200
70 X=INP(&HA8):POKE&HE201,X
80 X=X AND 240
90 X=X+X/16
100 POKE&HE200,X
110 :
120 REM * COPIEREN VAN "BOVENSTE" *
130 REM * 32K RAM IN HET ONGE-    *
140 REM * BRUIKTE 32K RAM         *
150 :
160 DATA F3         :' DI
```

```
170 DATA 3A,00,E2 :' LD A,(E200H)
180 DATA D3,A8    :' OUT (A8H),A
190 DATA 21,00,80 :' LD HL,8000H
200 DATA 11,00,00 :' LD DE,0000H
210 DATA 01,00,80 :' LD BC,8000H
220 DATA ED,B0    :' LDIR
230 DATA 3A,01,E2 :' LD A,(E201H)
240 DATA D3,A8    :' OUT (A8H),A
250 DATA FB       :' EI
260 DATA C9       :' RET
270 DATA "**"
280 :
290 T=&HE202
300 READ A$:IF A$="**" THEN310 ELSE POKE
 T,VAL("&H"+A$):T=T+1:GOTO300
310 T=&HE220
320 :
330 REM ****  COPIEER ROUTINE  ****
340 :
350 DATA F3        :'DI
360 DATA 33        :'INC SP
370 DATA 33        :'INC SP
380 DATA 23        :'INC HL
390 DATA CD,6C,00:'CALL 006CH
400 DATA F3        :'DI
410 DATA E5        :'PUSH HL
420 DATA 3A,00,E2:'LD A,(E200H)
430 DATA D3,A8     :'OUT (A8H),A
440 DATA 21,00,80:'LD HL,8000H
450 DATA 11,00,00:'LD DE,0000H
460 DATA C1        :'POP BC
470 DATA 1A        :'LD A,(DE):SPRONG
480 DATA F5        :'PUSH AF
490 DATA 7e        :'LD A,(HL)
500 DATA 12        :'LD (DE),A
510 DATA F1        :'POP AF
```

```
520 DATA 77       :'LD (HL),A
530 DATA 23       :'INC HL
540 DATA 13       :'INC DE
550 DATA 7A       :'LD A,D
560 DATA FE,80    :'CP 80H
570 DATA 20,F3    :'JR NZ,SPRONG
580 DATA 3A,01,E2:'LD A,(E201H)
590 DATA D3,A8    :'OUT (A8H),A
600 DATA E1       :'POP HL
610 DATA C5       :'PUSH BC
620 DATA FB       :'EI
630 DATA C3,28,41:'JP 4128
640 DATA "**"
650 :
660 RESTORE350
670 READ A$:IF A$="**" THEN680 ELSE POKE
 T,VAL("&H"+A$):T=T+1:GOTO670
680 POKE&HFE0D,&HC3:POKE&HFE0E,&H20
690 POKE&HFE0F,&HE2
700 DEFUSR=&HE202:X=USR(0)
710 CMD
```

Software matige RESET

De volgende routine voorziet elke MSX computer die niet standaard uitgerust is met een RESET-toets, alsnog met deze voorziening. Als u tijdens een programma (BASIC/Machinetaal) op de ESC-toets drukt dan wordt er een warme start uitgevoerd.

U dient dit programma in te laden voordat u ieder ander programma in laadt. Na het runnen is de ESC toets geactiveerd.

De routine kan bijvoorbeeld zijn nut bewijzen na het spelen van een spel. U hoeft dan nl. niet eerst uw computer uit te zetten voordat u een ander spel in gaat laden.

N.B.: De reset werkt alleen als bij het indrukken van de CAPS-LOCK-toets het lichtje aan of uit gaat. Gebeurt dit niet dan is het toetsenbord helaas afgeschakeld en zal een druk op de ESC-toets geen gevolg hebben.

```
1 REM ***************************
2 REM *                         *
3 REM *    SOFTWARE RESET VIA   *
4 REM *    DE ESC-TOETS         *
5 REM *                         *
6 REM ***************************
7 :
10 REM *****  DE MC-ROUTINE  *****
20 :
30 DATA FE,3A    :REM  CP 3AH
40 DATA CA,00,00:REM  JP Z,0000
50 DATA C9       :REM  RET
60 :
```

```
70 T=&HFFF6
80 FOR X=0 TO 5:READA$:POKE T+X,VAL("&H"
+A$):NEXT
90 POKE&HFDCD,&HF6:POKE&HFDCE,&HFF
100 POKE&HFDCC,&HC3
```

"OK"-prompt veranderen

MSX-2 computers bezitten het statement "SET PROMPT". Met dit kommando is het mogelijk om de "OK"-prompt te veranderen in iedere gewenste tekst. Dit gebeurt via SET PROMPT "promptnaam". Promptnaam mag echter niet langer zijn dan 6 karakters.

Onderstaande routine stelt de MSX-1 gebruiker ook in staat om de prompt te veranderen. Een groot voordeel ten opzichte van MSX-2 is dat de nieuwe prompt elke lengte (tot ongeveer 30) kan aannemen.

Het veranderen van de prompt gebeurt door aan de variabele A$ de gewenste alfanumerieke waarde toe te kennen.

```
1 REM ***************************
2 REM *                         *
3 REM * Ok veranderen in elke   *
4 REM *    gewenste tekst.      *
5 REM * Tekst definiëren in     *
6 REM * regel 120               *
7 REM *                         *
8 REM ***************************
9 :
100 CLEAR 200,&HE200
110 T=&HE200
120 A$="Truuks & Tips deel 3
130 POKET,34
140 FOR X=1TOLEN(A$):M$=MID$(A$,X,1):POK
E X+T,ASC(M$):NEXT:POKEX+T,0
150 :
160 REM ****  DE ROUTINE  ****
```

```
170 :
180 DATA 21,00,E2 :'LD HL,E200H
190 DATA E5       :'PUSH HL
200 DATA 2A,DC,F3 :'LD HL,(F3DCH)
210 DATA 2C       :'INC L
220 DATA CD,C6,00 :'CALL 00C6H
230 DATA 3E,0D    :'LD A,0DH
240 DATA CD,A2,00 :'CALL 00A2H
250 DATA E1       :'POP HL
260 DATA CD,24,4A :'CALL 4A24H
270 DATA C3,31,41 :'JP 4131H
280 :
290 T=&HE220
300 FOR X=0 TO22:READA$:POKE T+X,VAL("&H
"+A$):NEXT
310 POKE&HFF07,&HC3:POKE&HFF08,&H20:POKE
&HFF09,&HE2
```

LIST- & gebruikersbeveiliging

Soms is het wenselijk om uw kostbare programma te beveiligen zodat andere mensen, indien ze geen toestemming hebben, het niet kunnen listen of runnen.
De meeste listbeveiligingen werken alleen nadat het programma ingeladen en gerund is. De onderstaande listbeveiliging treedt echter al direkt in werking zodra het programma ingeladen is. Een beveiligd programma loopt alleen als het juiste password ingevoerd wordt.

De listbeveiliging valt uiteen in twee delen: Deel A en Deel B. Deel A bevat de machinetaal om een programma te beveiligen, deel B bevat de machinetaal en tekst om ervoor te zorgen dat het programma alleen verder gaat als het juiste password ingegeven wordt.

Welke handelingen moet u verrichten om een programma te beveiligen?

1. Laad deel A in en run het.
2. Laad het te beveiligen BASIC programma in de computer.
3. Plaats deel B van de beveiliging voor het te beveiligen BASIC programma. Neem alle regels zorgvuldig over! ! ! (de regelnummering hoeft niet het zelfde te zijn maar alle regels moeten er wel in staan).
4. Tik in: X=USR (0) + (ENTER/RETURN)
5. Schrijf het nu beveiligde programma eerst weg alvorens het te runnen.

U kunt het password "geheim" naar wens veranderen.

```
1 '********************************
2 '*          PROGRAMMA A          *
3 '*                               *
```

```
4 '*  MACHINETAAL ROUTINE OM        *
5 '*  EEN PROGRAMMA TEGEN LISTEN   *
6 '*  TE BEVEILGIGEN. HET PRO-     *
7 '*  GRAMMA WORDT AANGEROEPEN     *
8 '*  MET X=USR(0)                 *
9 '********************************
10 WIDTH40:T=&HE200
20 CLEAR 200,T:T=&HE200
30 FOR X=0 TO 7:READ A$:POKET,VAL("&H"+A
$):T=T+1:NEXT:DEFUSR=&HE200
40 DATA f3,21,ff,ff,22,03,80,c9
50 END
```

```
10 REM
20  '********************************
30  '*                              *
40  '*     PROGRAMMA BEVEILIGING    *
50  '*     M.B.V. MACHINETAAL       *
60  '*                              *
70  '*        PROGRAMMA B           *
80  '********************************
90  :
100 '********************************
110 '* REGELS 200-220 BEVATTEN DE    *
120 '* MACHINETAAL OM HET PROGRAMMA, *
130 '* NA EEN GOED PASSWORD, WEER    *
140 '* IN ORDE TE MAKEN. DE ROUTINE  *
150 '* WORDT AANGEROEPEN MET X=USR(0)*
160 '********************************
170 :
180  CLEAR 200,&HE200:T=&HE200:DEFUSR=T
190  FOR X=0 TO 7:READ A$:POKET,VAL("&H"
+A$):T=T+1:NEXT
200  DATA f3,21,00,00,22,03,80,c9
210 :
220 SCREEN0:WIDTH40
```

```
230 PRINT "Dit programma kan noch gelist
, noch ge- rund worden als u het juiste
password  niet kent."
240 PRINT:INPUT"Geef het password";A$
250 IF A$<>"geheim" THEN PRINT "Verkeerd
 password, u hebt nog 1 kans.
260 IF A$="geheim" THEN CLS:X=USR(0):GOT
O300
270 PRINT:INPUT"Geef het password";A$
280 IF A$<>"geheim" THEN PRINT"Helaas, h
et  password is weer verkeerd dit progr
amma staat niet meer in het ge-heugen.":
NEW
290 X=USR(0)
300 :
310 PRINT"U heeft het juiste password in
gevoerd":PRINT"U kunt dit programma nu g
ebruiken."
```

Twee schermen uitwisselen onder SCREEN2 of 3

De volgende machinetaal routine stelt u in staat om twee grafische schermen uit te wisselen. Het programma is alleen geschikt voor 64K MSX computers.

Werking van routine

Nadat u dit programma gerund heeft, bent u in staat om met het kommando "X=USR (0)" twee grafische schermen onderling uit te wisselen. Ook de eventuele aanwezige sprites worden meegenomen.
U dient de volgende handelingen te verrichten.

1. Maak een programma dat een tekening in screenmode 2 of 3 maakt (eerste tekening).
2. Voeg aan het eind van dit programma de volgende BASIC statements toe "X=USR (0): SCREEN 2/3." De tekening is nu opgeslagen in het RAM.
3. Maak nog een programma dat een tekening tekent (tweede tekening).
4. Als u het kommando "X=USR (0)" aan het einde van dit programma zet, dan worden de eerste en tweede tekening verwisseld. De eerste tekening komt in het VRAM, de tweede in het RAM.
5. Als u nogmaals het kommando "X=USR (0)" geeft, dan worden de tekeningen weer verwisseld.

Om het geheel wat duidelijker te maken is in onderstaand programma een voorbeeld verwerkt.

```
1 REM ***************************
2 REM *                         *
```

```
3 REM * SCHERMEN WISSELEN D.M.V.*
4 REM *                         *
5 REM * EEN MACHINETAAL ROU-    *
6 REM *                         *
7 REM * TINE                    *
8 REM *                         *
9 REM ***************************
10 GOTO 310
20 :
30 REM ***************************
40 REM *                         *
50 REM * TEKST: MSX              *
60 REM *                         *
70 REM ***************************
80 :
90 SCREEN3
100 LINE(40,40)-(210,130),8,BF
110 LINE(50,50)-(200,120),7,BF
120 COLOR 1:CLOSE:OPEN"grp:"AS1:PSET(80,
70),POINT(80,70):PRINT#1,"MSX"
130 :
140 X=USR(0):SCREEN3
150 :
160 REM ***************************
170 REM *                         *
180 REM * TEKST: MSX TRUUKS &     *
190 REM *        TIPS DEEL 3      *
200 REM *                         *
210 REM ***************************
220 :
230 LINE(10,20)-(230,190),11,BF
240 LINE(20,30)-(220,180),9,BF
250 PSET(30,40):PRINT#1,"TRUUKS":PSET(30
,90):PRINT#1,"& TIPS":PSET(30,140):PRINT
#1,"DEEL 3
260 :
```

```
270 X=USR(0)
280 X=USR(0)
290 FOR M=1 TO 300:NEXT:GOTO 280
300 :
310 CLEAR 200,&HE200
320 X=INP(&HA8):POKE&HE201,X
330 X=X AND 240
340 X=X+X/16
350 POKE&HE200,X
360 :
370 DATA F3          :' DI
380 DATA 21,00,00    :' LD HL ,0000
390 DATA CD,4A,0     :' LOOP:CALL 004A
400 DATA f3          :' DI
410 DATA F5          :' PUSH AF
420 DATA 3A,00,E2    :' LD A,(E200H)
430 DATA D3,A8       :' OUT (A8H),A
440 DATA 46          :' LD B,(HL)
450 DATA F1          :' POP AF
460 DATA 77          :' LD (HL),A
470 DATA C5          :' PUSH BC
480 DATA 3A,01,E2    :' LD A,(E201H)
490 DATA D3,A8       :' OUT (A8H),A
500 DATA F1          :' POP AF
510 DATA CD,4D,00    :' CALL 004D
520 DATA F3          :' DI
530 DATA 23          :' INC HL
540 DATA 7C          :' LD A,H
550 DATA FE,40       :' CP 40
560 DATA 20,E2       :' JP NZ,LOOP
570 DATA FB          :' EI
580 DATA C9          :' RET
590 DATA "**"
600 T=&HE202
610 READ A$:IF A$="**" THEN 620 ELSE POK
E T,VAL("&H"+A$):T=T+1:GOTO610
```

```
620 DEFUSR=&HE202
630 GOTO 80
```

Aantekeningen

Aantekeningen

Aantekeningen

Aantekeningen

Aantekeningen

Aantekeningen

Nederlandstalige MSX handboeken

MSX BASIC handboek voor iedereen, door A.C.J. Groeneveld
Een kompleet nederlandstalig handboek voor iedere MSX computergebruiker. Dit handboek omvat een volledige behandeling van het MSX-basic in het Nederlands. Het handboek geeft een antwoord op elke vraag die een programmeur, van welke scholing ook, over het MSX-basic zou kunnen stellen. De volledige syntaxisbehandeling rekent af met onzekerheden of een bepaalde schrijfwijze nu wel of niet is toegestaan. De duidelijke beschrijving geeft per sleutelwoord aan, welke de funktie hiervan is. De laatste mogelijk nog aanwezig onduidelijkheden worden vervolgens door de opgenomen, zinvolle voorbeelden weggenomen ISBN 90 6398 1007

MSX ZAKBOEKJE door Wessel Akkermans
Een vlot geschreven naslagwerk na of naast het handboek. U vindt er o.a. in: niet computergerichte tabellen; de MSX-BASIC instructieset; diverse tabellen die het BASIC-programmeren kunnen versnellen; de Z80 instructieset; hardware-gegevens (connectoren) en een aantal programmaatjes ISBN 90 6398 888 5

MSX DISK handboek voor iedereen, door A.C.J. Groeneveld
Handboek voor diskdrivebezitters om naast het grote handboek te gebruiken. Een zeer volledige behandeling van het disk-gebeuren zelf en de specifieke disk kommando's, uitgebreid met voorbeelden, tabellen en overzichten. Het handboek is aangevuld met interessante programma's, waaronder een tekentafelprogramma en een basisprogramma voor basisonderhoud ISBN 90 6398 407 3

MSX PRAKTIJKPROGRAMMA'S door Wessel Akkermans
Praktische programma's met waar nodig eerst een stukje theorie. Erg handig bij het maken van uw programma's. Een greep uit de onderwerpen: priemgetallen; zoeken en sorteren; trefwoordenlijsten; converteren van getallen; enz. ISBN 90 6398 437 5

MSX QUICK DISK handboek voor iedereen, door A.C.J. Groeneveld
Hèt handboek voor iedere QUICK DISK gebruiker. Uitvoerige behandeling van de sleutelwoorden aangevuld met duidelijke voorbeelden met listing ISBN 90 6398 254 2

MSX DOS handboek voor iedereen, door A.C.J. Groeneveld
Dit handboek geeft u op een heldere wijze een totaalbeeld van de mogelijkheden van het MSX-DOS. Ook is dit handboek voorzien van een inleiding op het begrip 'operating system' en dus echt een handboek voor iedereen ISBN 90 6398 674 2

MSX LEERBOEKEN
door Wessel Akkermans en Piet den Heijer
De serie MSX leerboeken geeft een komplete kursus MSX-BASIC programmeren, in drie delen. Deze leerboeken zijn gericht op de beginnende programmeur. De moeilijkheidsgraad van de leerstof wordt dan ook slechts geleidelijk hoger. De gebruikte voorbeelden zijn zo praktisch mogelijk gekozen. Hierdoor kunnen al in een vroeg stadium bruikbare programma's worden gemaakt. Dit zal de lezer/leerling er toe aansporen om verder te gaan. Aan het eind van ieder deel is een groot voorbeeldprogramma opgenomen. Dit programma laat zien waartoe de lezer/leerling na bestudering van het betreffende leerboek in staat zal zijn.
Bij ieder leerboek is een afzonderlijk –Opdrachten en uitwerkingen– boekje te verkrijgen. In deze boekjes staan, in volgorde van de hoofdstukken uit het leerboek, vragen en opdrachten met antwoorden en uitwerkingen. Een unieke serie leerboeken voor een ieder die meer over MSX wil weten en het betere werk met zijn computer wil maken.

MSX Basic leerboek deel 1 - ISBN 90 6398 649 1
Opdrachten bij deel 1 - ISBN 90 6398 596 7
MSX Basic leerboek deel 2 - ISBN 90 6398 769 2
Opdrachten bij deel 2 - ISBN 90 6398 556 8
MSX DOS leerboek deel 3 - ISBN 90 6398 519 3
Opdrachten bij deel 3 - ISBN 90 6398 516 9

MSX Verder uitgediept door H. Klopper
Eindelijk een Nederlandstalig boek over het altijd in de mist gehulde onderwerp – PEEKS EN POKES. In dit boek staan alle belangrijke RAM en VRAM adressen. De video chip en zijn registers worden volledig uitgelegd. Maar ook hoe men een machinetaal programma van cassette naar disk kan schrijven. Bovendien een diskloader utility en een uiterst geavanceerde programma beveiliging. Tenslotte zijn er een aantal interessante programma's opgenomen, waaronder een wereldkaart, waarmee verder kan worden geëxperimenteerd. Elke MSX gebruiker kan in dit boek iets van zijn gading vinden en nieuws leren.

ISBN 90 6398 447 2

MSX Machinetaal handboek door H. Klopper en M. Le Belle
Hoewel een MSX computer over een krachtig Basic beschikt, is het toch handig tijdens het programmeren de grondbeginselen van machinetaal te kennen. Daarvoor is dit boek een goede gids. De zaken worden niet puur theoretisch maar ook aan de hand van duidelijke voorbeelden, die direkt bruikbaar zijn, uitvoerig uitgelegd. Enkele onderwerpen zijn verder – scroll routine –machinetaal software (ook in disk Basic) op casette zetten –disassembler –Z80 essembler instrukties –lijst van ROM-routines –alle hook-adressen –bespreking van Basic tokens en een kompleet token-overzicht. Hèt handboek voor iedere MSX programmeur die zijn computer ten volle wil benutten.

ISBN 90 6398 735 8

MSX TRUUKS EN TIPS
door A.C.J. Groeneveld
Hoe laat ik de computer een cirkel arceren, hoe tover ik mijn computer om in een elektronisch orgeltje, hoe maak ik een mooie intro voor een spelletje. Allemaal vraagstukken die zich lastig laten programmeren maar die iedere MSX-er toch graag opgelost wil zien.
Dit boekje staat boordevol truuks en tips, allemaal in gewoon MSX basic geschreven. Bladerend door dit boek komt u tot de ontdekking dat er voornamelijk korte maar uiterst krachtige en bijzonder goed bruikbare routines zijn opgenomen. Dit boekje geeft kort maar krachtig een antwoord op al uw programmeervragen.

deel 1 ISBN 90 6398 900 8
deel 2 ISBN 90 6398 340 9

SOFTWARE PLUS IN MSX

INTROTAPE MSX door A.C.J. Groeneveld
Heeft u nog maar net een MSX computer gekocht en wilt u graag weten wat de computer kan en hoe u hem kunt leren programmeren? Deze cassette introduceert MSX op een uiterst vriendelijke en onderwijzende manier. U krijgt instructies hoe u de computer aan moet sluiten en de tape laden. Daarna volgt een demonstratie van de mogelijkheden in MSX, zoals het tekenen van sprites en het werken met de driestemmige toongenerator. Het geheel wordt afgesloten met twee 'les' gedeeltes. In anderhalf à drie uur weet u wat de MSX computer is, wat hij kan, en heeft u haast ongemerkt al wat regels geprogrammeerd.

ISBN 90 6398 148 1

MSX SCRIPT door Ton Weijters
Een menugestuurde nederlandstalige tekstverwerker. Het programma is geschikt om efficiënt grotere of kleinere teksten te bewerken. Pagina-indeling (regellengte, paginalengte, marge, inspringen, centreren, enz.) wordt door het programma verzorgd. Dit geldt ook voor de paginatelling, toptitel en het eventueel uitvullen van de regels. Ook corrigeren, zoeken, string-substitutie, blokken tekst verplaatsen, copiëren of verwijderen, onderstrepen en vet zetten, is mogelijk met dit programma.

ISBN 90 6398 189 9

MSX DRAWS door A.C.J. Groeneveld
Een tekenprogramma in MSX basic, waarmee u al binnen 10 minuten uw eerste tekening kunt maken. Draws werkt erg vriendelijk en maakt gebruik van alle grafische mogelijkheden van de MSX computer. U kunt met Draws zowel technisch als creatieve tekeningen maken. Het programma heeft een effectief bereik van ruim 30.000 bij 30.000 puntjes met mogelijkheden als lijnen, cirkels, krommen, inkleuren, vergroten, verkleinen, verschuiven, verdraaien en andere tekeningen invoegen

ISBN 90 6398 754 4

truuks en tips
deel 3

De MSX programmeur heeft de beschikking over een ruim scala van mogelijkheden. Deze mogelijkheden zijn echter niet zomaar om te zetten in een goed programma. De MSX truuks en tips reeks heeft als doel u in dit opzicht een ruggesteuntje te bieden.
In navolging op de eerste twee delen, kunt u in dit boekje weer de nodige nuttige routines en programma's terug vinden die u nergens anders zult aantreffen. De trukendoos wordt in dit deel nog verder opengegooid. Naast programma's geschreven in Basic komen nu ook enkele razend snelle machinetaalroutines op de proppen. Een kleine selektie uit de inhoud:

- Uitwisselen van twee grafische schermen (16k VRAM)
- Geavanceerde listbeveiligingen
- Verschillende series funktietoetsen die u direkt kunt oproepen
- Programma's automatisch starten na een cload instruktie
- Bankswitch routine, waardoor u twee banken van 24k tot uw beschikking heeft. U kunt zelfs twee Basic programma's tegelijk in het geheugen bewaren
- De 'Ok-prompt' veranderen in elke gewenste tekst
- Sprite detektie (kijken welke sprites hebben gebotst).

Dit is slechts een kleine greep uit de vele uitgekookte truuks en tips die ook dit deel weer bevat.
MSX truuks en tips deel 3 geeft u meer dan ooit het gevoel de computer volledig te beheersen.

ISBN 906398 910 5